

BBBBBBBB	AAAAAA	SSSSSS	CCCCCCC	TTTTTTT	RRRRRRR	LL	000000	000000		
BBBBBBBB	AAAAAA	SSSSSS	CCCCCCC	TTTTTTT	RRRRRRR	LL	00	00		
BB	BB	AA	AA	SS	CC	TT	RR	RR		
BB	BB	AA	AA	SS	CC	TT	RR	RR		
BB	BB	AA	AA	SS	CC	TT	RR	RR		
BB	BB	AA	AA	SS	CC	TT	RR	RR		
BBBBBBBB	AA	AA	SSSSS	CC	TT	RRRRRRR	LL	00	00	
BBBBBBBB	AA	AA	SSSSS	CC	TT	RRRRRRR	LL	00	00	
BB	BB	AAAAAA	SS	CC	TT	RR	RR	00	00	
BB	BB	AAAAAA	SS	CC	TT	RR	RR	00	00	
BB	BB	AA	AA	SS	CC	TT	RR	RR	00	00
BB	BB	AA	AA	SS	CC	TT	RR	RR	00	00
BBBBBBBB	AA	AA	SSSSSS	CCCCCCC	TT	RR	RR	LL	000000	000000
BBBBBBBB	AA	AA	SSSSSS	CCCCCCC	TT	RR	RR	LL	000000	000000

LL		SSSSSSS
LL		SSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSS
LL		SSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSSS
LL		SSSSSSS

```
1 0001 0 MODULE BAS$TRL0 (
2 0002 0           IDENT = '1-004'
3 0003 0           ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1 !
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: VAX-11 BASIC Miscellaneous I/O
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the BASIC CTRLO and RCTRLO functions,
36 0036 1 Which suppress and unsuppress output on a specified channel.
37 0037 1
38 0038 1 ENVIRONMENT: VAX-11 User Mode
39 0039 1
40 0040 1 AUTHOR: John Sauter, CREATION DATE: 19-APR-1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original.
45 0045 1 1-002 - Set up ISBSA_USER FP. JBS 25-JUL-1979
46 0046 1 1-003 - Correct test of LOBSV OPENED. JBS 26-FEB-1980
47 0047 1 1-004 - Set CCO bit on the output side of channel 0. JBS 31-MAR-1980
48 0048 1 --
49 0049 1
50 0050 1 !<BLF/PAGE>
```

```
52 0051 1 | SWITCHES:  
53 0052 1 |  
54 0053 1 |  
55 0054 1 |  
56 0055 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);  
57 0056 1 |  
58 0057 1 |  
59 0058 1 |  
60 0059 1 |  
61 0060 1 |  
62 0061 1 | REQUIRE 'RTLIN:OTSLNK';  
63 0490 1 | ! Define linkages  
64 0491 1 |  
65 0492 1 |  
66 0493 1 |  
67 0494 1 |  
68 0495 1 | FORWARD ROUTINE  
69 0496 1 | BASSCTRL0,  
70 0497 1 | BASSRCTRL0;  
71 0498 1 |  
72 0499 1 |  
73 0500 1 |  
74 0501 1 |  
75 0502 1 |  
76 0503 1 | REQUIRE 'RTLML:OTSLUB';  
77 0643 1 | ! Get LUB definitions  
78 0644 1 | REQUIRE 'RTLML:OTYSISB';  
79 0812 1 | ! Get ISB definitions  
80 0813 1 | REQUIRE 'RTLIN:RTLPSECT';  
81 0908 1 | ! Macros for defining psects  
82 0909 1 | LIBRARY 'RTLSTARLE';  
83 0910 1 | ! System symbols  
84 0911 1 |  
85 0912 1 |  
86 0913 1 |  
87 0914 1 |  
88 0915 1 |  
89 0916 1 |  
90 0917 1 |  
91 0918 1 |  
92 0919 1 |  
93 0920 1 |  
94 0921 1 |  
95 0922 1 |  
96 0923 1 |  
97 0924 1 |  
98 0925 1 |  
99 0926 1 |  
100 0927 1 |  
101 0928 1 |  
102 0929 1 |  
103 0930 1 |  
104 0931 1 |  
105 0932 1 |  
106 0933 1 |  
107 0934 1 |  
108 0935 1 |  
| EXTERNAL ROUTINE  
| BASS$OPEN ZERO : NOVALUE,  
| BASS$CB_PUSH : JSB CB PUSH NOVALUE,  
| BASS$CB_POP : JSB CB POP NOVALUE,  
| BASS$STOP_10 : NOVALUE;  
| ! Open channel zero  
| ! Load register CCB  
| ! Done with register CCB  
| ! Signal fatal I/O error
```

```
: 109      0936 1
: 110      0937 1
: 111      0938 1  + The following are the error codes used in this module.
: 112      0939 1  -
: 113      0940 1
: 114      0941 1  EXTERNAL LITERAL
: 115      0942 1  BAS$K_IO_CHANOT : UNSIGNED (8);      ! Channel not open.
: 116      0943 1
```

```
118 0944 1 GLOBAL ROUTINE BAS$CTRL0 (           ! Suppress output
119 0945 1   (CHAN                         ! Channel on which to suppress output
120 0946 1   ) =
121 0947 1
122 0948 1 ++
123 0949 1   FUNCTIONAL DESCRIPTION:
124 0950 1
125 0951 1   Simulates typing a control 0 on the terminal open on the
126 0952 1   specified channel.
127 0953 1
128 0954 1   FORMAL PARAMETERS:
129 0955 1
130 0956 1   CHAN.rl.v      The channel whose terminal to simulate a
131 0957 1   control 0 on
132 0958 1
133 0959 1   IMPLICIT INPUTS:
134 0960 1   NONE
135 0961 1
136 0962 1   IMPLICIT OUTPUTS:
137 0963 1
138 0964 1   LUB$V_CCO      Cancel control 0.
139 0965 1
140 0966 1   ROUTINE VALUE:
141 0967 1   COMPLETION CODES:
142 0968 1
143 0969 1   0970 1   SSS_NORMAL
144 0971 1
145 0972 1   SIDE EFFECTS:
146 0973 1
147 0974 1   Signals if an error is encountered.
148 0975 1   BAS$SCB_PUSH will signal if the channel number is invalid.
149 0976 1   This function is a no-operation if the channel is not open.
150 0977 1
151 0978 1   !--
152 0979 1
153 0980 2   BEGIN
154 0981 2
155 0982 2   BUILTIN
156 0983 2   FP;
157 0984 2
158 0985 2   GLOBAL REGISTER
159 0986 2   CCB = K_CCB_REG : REF BLOCK [, BYTE];
160 0987 2
161 0988 2   LOCAL
162 0989 2   FMP : REF BLOCK [, BYTE];
163 0990 2
164 0991 2   FMP = .FP;
165 0992 2
166 0993 2   + Get the CCB for the channel.
167 0994 2   -
168 0995 2
169 0996 3   IF (.CHAN EQL 0)
170 0997 3   THEN
171 0998 3   BEGIN
172 0999 3
173 1000 3   + The user is referencing his controlling terminal.
```

```

175      1001      !-
176      1002      BASS$CB_PUSH (LUB$K_LUN_BPRI, LUB$K_ILUN_MIN);
177      1003      [CB [ISBSA_USER_FP] = .FMP [SF$L_SAVE_FP];
178      1004      !+
179      1005      If the controlling terminal is not yet open, open it.
180      1006      !-
181      1007      IF ( NOT .[CB [LUB$V_OPENED]) THEN BASS$OPEN_ZERO (.FMP [SF$L_SAVE_FP]);
182      1008      !+
183      1009      ELSE END
184      1010      ELSE
185      1011      BEGIN
186      1012      !+
187      1013      This is an ordinary channel.
188      1014      !-
189      1015      BASS$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
190      1016      [CB [ISBSA_USER_FP] = .FMP [SF$L_SAVE_FP];
191      1017      END;
192      1018      !-
193      1019      !+
194      1020      If the channel is not now open, this function is a no-operation.
195      1021      !-
196      1022      !+
197      1023      !+
198      1024      IF (.CB [LUB$V_OPENED])
199      1025      THEN
200      1026      BEGIN
201      1027      !+
202      1028      Now clear the CCO bit, so control 0's will not be canceled.
203      1029      !-
204      1030      [CB [LUB$V_CCO] = 0;
205      1031      END;
206      1032      !-
207      1033      !+
208      1034      We are done with register CCB.
209      1035      !-
210      1036      BASS$CB_POP ();
211      1037      RETURN TSS$NORMAL;
212      1038      END;                                ! end of BASSCTRL0

```

```

.TITLE BASSCTRL0
.IDENT \1-004\

.EXTRN BASS$OPEN_ZERO, BASS$CB_PUSH
.EXTRN BASS$CB_POP, BASS$STOP_IO
.EXTRN BASS$K_ID_CHANOT

.PSECT _BASS$CODE,NOWRT, SHR, PIC,2

```

54 00000000G	00 9E 00002	.ENTRY BASSCTRL0, Save R2,R3,R4,R11	: 0944
53	5D D0 00009	MOVAB BASS\$CB_PUSH, R4	: 0991
	04 AC D5 0000C	MOVL FP, FMP	: 0996
50	1E 12 0000F	TSTL CHAN	1002
52	08 CE 00011	BNEQ 1\$	
	08 CE 00014	MNEGL #8, R0	
	64 16 00017	MNEGL #8, R2	
		JSB BASS\$CB_PUSH	

FF4C	CB	0C	A3	DD	00019	MOVL	12(FMP), -180(CCB)	: 1003	
	1E	FC	AB	E8	0001F	BLBS	-4(CCB), 3\$: 1008	
		0C	A3	DD	00023	PUSHL	12(FMP)		
00000000G	00		01	FB	00026	CALLS	#1, BASS\$OPEN_ZERO		
			OE	11	0002D	BRB	2\$		
			50	D4	0002F	1\$:	CLRL	: 0996	
	52	04	AC	DD	00031	MOVL	CHAN, R2	: 1016	
				64	16	00035	JSB	BASS\$CB_PUSH	
FF4C	CB	0C	A3	DD	00037	MOVL	12(FMP), -180(CCB)	: 1017	
	04	FC	AB	E9	0003D	2\$:	BLBC	-4(CCB), 4\$: 1024
A0	AB	04	8A	00041	3\$:	BICB2	#4, -96(CCB)	: 1030	
		00000000G	00	16	00045	4\$:	JSB	BASS\$CB_POP	: 1036
	50		01	DD	00048	MOVL	#1, R0	: 1037	
					04	0004E	RET	: 1038	

: Routine Size: 79 bytes, Routine Base: _BASS\$CODE + 0000

: 213 1039 1

```
1040 1 GLOBAL ROUTINE BAS$CTRL0 (           ! Cancel a typed control 0
1041 1     CHAN                         ! Channel on which to do this
1042 1     ) =
1043 1
1044 1     ++
1045 1     FUNCTIONAL DESCRIPTION:
1046 1
1047 1     Cancels control 0 on the terminal open on the specified channel.
1048 1
1049 1     FORMAL PARAMETERS:
1050 1
1051 1     CHAN.rl.v      The channel whose terminal to disable CTRL0ing on
1052 1
1053 1     IMPLICIT INPUTS:
1054 1
1055 1     NONE
1056 1
1057 1     IMPLICIT OUTPUTS:
1058 1
1059 1     LUB$V_CCO which, when set, cancels control 0.
1060 1
1061 1     ROUTINE VALUE:
1062 1     COMPLETION CODES:
1063 1
1064 1     SSS_NORMAL
1065 1
1066 1     SIDE EFFECTS:
1067 1
1068 1     Signals if an error is encountered.
1069 1     BAS$$_CB_PUSH will signal if the channel number is invalid.
1070 1     This routine is a no-operation if the channel is not open.
1071 1
1072 1     --
1073 1
1074 2     BEGIN
1075 2
1076 2     BUILTIN
1077 2     FP;
1078 2
1079 2     GLOBAL REGISTER
1080 2     CCB = K_CCB_REG : REF BLOCK [, BYTE];
1081 2
1082 2     LOCAL
1083 2     FMP : REF BLOCK [, BYTE];
1084 2
1085 2     FMP = .FP;
1086 2
1087 2     + Get the CCB for the channel.
1088 2     -
1089 2
1090 3     IF (.CHAN EQ 0)
1091 2     THEN
1092 3     BEGIN
1093 3
1094 3     + The user is referencing his controlling terminal.
1095 3     -
1096 3     BAS$$_CB_PUSH (LUB$K_LUN_BPRI, LUB$K_ILUN_MIN);
```

```

272 1097 3 CCB [ISBSA_USER_FP] = .FMP [SF$L_SAVE_FP];
273 1098 3
274 1099 3 |+ If the controlling terminal is not yet open, open it.
275 1100 3 |-
276 1101 3
277 1102 3 |+ IF ( NOT .CCB [LUB$V_OPENED] ) THEN BASS$OPEN_ZERO (.FMP [SF$L_SAVE_FP]);
278 1103 3
279 1104 3 |+ END
280 1105 2 |+ ELSE
281 1106 3 |+ BEGIN
282 1107 3
283 1108 3 |+ This is an ordinary channel.
284 1109 3
285 1110 3 |+ BASS$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
286 1111 3 |+ CCB [ISBSA_USER_FP] = .FMP [SF$L_SAVE_FP];
287 1112 2 |+ END;
288 1113 2
289 1114 2 |+ If the channel is not now open, this function is a no-operation.
290 1115 2
291 1116 2
292 1117 2
293 1118 3 |+ IF (.CCB [LUB$V_OPENED])
294 1119 2 |+ THEN
295 1120 3 |+ BEGIN
296 1121 3
297 1122 3 |+ Now set the CCO bit, which will cause the record level code
298 1123 3 |+ to tell RMS to cancel control 0.
299 1124 3
300 1125 3 |+ CCB [LUB$V_CCO] = 1;
301 1126 2 |+ END;
302 1127 2
303 1128 2 |+ We are done with register CCB.
304 1129 2 |-
305 1130 2
306 1131 2 |+ BASS$CB_POP ();
307 1132 2 |+ RETURN TSSS_NORMAL;
308 1133 1 |+ END;

```

54	00000000G	00	9E	00002	.ENTRY	BASSRCTRL0, Save R2,R3,R4,R11						1040
53		04	00	00009	MOVAB	BASS\$CB_PUSH, R4						1085
			04	AC D5 0000C	MOVL	FP, FMP						1090
				1E 12 0000F	ISTL	CHAN						
50				08 CE 00011	BNEQ	1\$						1096
52				08 CE 00014	MNEGL	#8, R0						
				64 16 00017	MNEGL	#8, R2						
FF4C	CB	0C	A3	00 00019	JSB	BASS\$CB_PUSH						1097
1E		FC	AB	E8 0001F	MOVL	12(FMP), -180((CB))						1102
		0C	A3	DD 00023	BLBS	-4((CB)), 3\$						
00000000G	00		01	FB 00026	PUSHL	12(FMP)						
			0E	11 0002D	CALLS	#1, BASS\$OPEN_ZERO						1090
			52	04 0002F 1\$:	BRB	2\$						1110
				50 D4 00031	CLRL	R0						
					MOVL	CHAN, R2						

FF4C	CB	0C	64	16	00035	JSB	BASS\$CB_PUSH	:	1111	
	04		A3	D0	00037	MOVL	12(FMP), -180(CCB)	:	1118	
A0	AB	FC	AB	E9	0003D	BLBC	-4(CCB), 4\$:	1125	
			04	88	00041	BISB2	#4, -96(CCB)	:	1131	
			00	00	16	JSB	BASS\$CB_POP	:	1132	
50				01	D0	00045	MOVL	#1, R0	:	1133
					04	0004B	RET			
						04	0004E			

: Routine Size: 79 bytes, Routine Base: _BASS\$CODE + 004F

309	1134	1							
310	1135	1	END						
311	1136	1							
312	1137	0	ELUDOM						

! end of module BASS\$CTRL0

PSECT SUMMARY

Name	Bytes	Attributes
_BASS\$CODE	158	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
\$_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	2	0	581	00:01.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:BASCTRL0/OBJ=OBJ\$:BASCTRL0 MSRC\$:BASCTRL0/UPDATE=(ENH\$:BASCTRL0)

: Size:	158 code + 0 data bytes
: Run Time:	00:10.0
: Elapsed Time:	00:24.3
: Lines/CPU Min:	6828
: Lexemes/CPU-Min:	40438
: Memory Used:	117 pages
: Compilation Complete	

0020 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASCLOSE
LIS

BASCONCAT
LIS

BASCUTRO
LIS

BASCHANGE
LIS

BASCRLC
LIS

BASCHAIN
LIS

BASECOPYFD
LIS

BASCHR
LIS

BASEMPAPP
LIS

BASCUTOUT
LIS

BASCPOS
LIS